Tabla de Códigos 1/2

V 2.3ES Traducido por Lucas Violini, Argentina

TRANSFERENCIA				Flags								
Nombre	**Comentario**	**Código**	**Operación**	**O**	**D**	**I**	**T**	**S**	**Z**	**A**	**P**	**C**
MOV	Mover (copiar)	MOV Dest,Fuente	Dest:=Fuente									
XCHG	Intercambiar	XCHG Op1,Op2	Op1:=Op2 , Op2:=Op1									
STC	Set the carry (Carry = 1)	STC	CF:=1									1
CLC	Clear Carry (Carry = 0)	CLC	CF:=0									0
CMC	Complementar Carry	CMC	CF:=Ø CF									±
STD	Setear dirección	STD	DF:=1 (interpreta strings de arriba hacia abajo)		1							
CLD	Limpiar dirección	CLD	DF:=0 (interpreta strings de abajo hacia arriba)		0							
STI	Flag de Interrupción en 1	STI	IF:=1			1						
CLI	Flag de Interrupción en 0	CLI	IF:=0			0						
PUSH	Apilar en la pila	PUSH Fuente	DEC SP, [SP]:=Fuente									
PUSHF	Apila los flags	PUSHF	O, D, I, T, S, Z, A, P, C 286+: También NT, IOPL									
PUSHA	Apila los registros generales	PUSHA	AX, CX, DX, BX, SP, BP, SI, DI									
POP	Desapila de la pila	POP Dest	Destino:=[SP], INC SP									
POPF	Desapila a los flags	POPF	O, D, I, T, S, Z, A, P, C 286+: También NT, IOPL	±	±	±	±	±	±	±	±	±
POPA	Desapila a los reg. general.	POPA	DI, SI, BP, SP, BX, DX, CX, AX									
CBW	Convertir Byte a Word	CBW	AX:=AL (con signo)									
CWD	Convertir Word a Doble	CWD	DX:AX:=AX (con signo)	±				±	±	±	±	±
CWDE	Conv. Word a Doble Exten.	CWDE 386	EAX:=AX (con signo)									
IN *i*	Entrada	IN Dest,Puerto	AL/AX/EAX := byte/word/double del puerto especifi.									
OUT *i*	Salida	OUT Puerto,Fuente	Byte/word/double del puerto especifi. := AL/AX/EAX									

i para más información ver especificaciones de la intrucción   Flags: ±=Afectado por esta instrucción ?=Indefinido luego de esta instrucción

ARITMÉTICOS				Flags								
Nombre	**Comentario**	**Código**	**Operación**	**O**	**D**	**I**	**T**	**S**	**Z**	**A**	**P**	**C**
ADD	Suma	ADD Dest,Fuente	Dest:=Dest+ Fuente	±				±	±	±	±	±
ADC	Suma con acarreo	ADC Dest,Fuente	Dest:=Dest+ Fuente +CF	±				±	±	±	±	±
SUB	Resta	SUB Dest,Fuente	Dest:=Dest- Fuente	±				±	±	±	±	±
SBB	Resta con acarreo	SBB Dest,Fuente	Dest:=Dest-(Fuente +CF)	±				±	±	±	±	±
DIV	División (sin signo)	DIV Op	Op=byte: AL:=AX / Op AH:=Resto	?				?	?	?	?	?
DIV	División (sin signo)	DIV Op	Op=word: AX:=DX:AX / Op DX:=Resto	?				?	?	?	?	?
DIV 386	División (sin signo)	DIV Op	Op=doublew.: EAX:=EDX:EAX / Op EDX:=Resto	?				?	?	?	?	?
IDIV	División entera con signo	IDIV Op	Op=byte: AL:=AX / Op AH:=Resto	?				?	?	?	?	?
IDIV	División entera con signo	IDIV Op	Op=word: AX:=DX:AX / Op DX:=Resto	?				?	?	?	?	?
IDIV 386	División entera con signo	IDIV Op	Op=doublew.: EAX:=EDX:EAX / Op EDX:=Resto	?				?	?	?	?	?
MUL	Multiplicación (sin signo)	MUL Op	Op=byte: AX:=AL*Op si AH=0 ♦	±				?	?	?	?	±
MUL	Multiplicación (sin signo)	MUL Op	Op=word: DX:AX:=AX*Op si DX=0 ♦	±				?	?	?	?	±
MUL 386	Multiplicación (sin signo)	MUL Op	Op=double: EDX:EAX:=EAX*Op si EDX=0 ♦	±				?	?	?	?	±
IMUL *i*	Multiplic. entera con signo	IMUL Op	Op=byte: AX:=AL*Op si AL es suficiente ♦	±				?	?	?	?	±
IMUL	Multiplic. entera con signo	IMUL Op	Op=word: DX:AX:=AX*Op si AX es suficiente ♦	±				?	?	?	?	±
IMUL 386	Multiplic. entera con signo	IMUL Op	Op=double: EDX:EAX:=EAX*Op si EAX es sufi. ♦	±				?	?	?	?	±
INC	Incrementar	INC Op	Op:=Op+1 (El Carry no resulta afectado !)	±				±	±	±	±	
DEC	Decrementar	DEC Op	Op:=Op-1 (El Carry no resulta afectado !)	±				±	±	±	±	
CMP	Comparar	CMP Op1,Op2	Op1-Op2	±				±	±	±	±	±
SAL	Desplazam. aritm. a la izq.	SAL Op,Cantidad	CF ◄ MSB … LSB ◄ 0	*i*				±	±	?	±	±
SAR	Desplazam. aritm. a la der.	SAR Op,Cantidad	► MSB … LSB ► CF	*i*				±	±	?	±	±
RCL	Rotar a la izq. c/acarreo	RCL Op,Cantidad	CF ◄ MSB … LSB ◄	*i*								±
RCR	Rotar a la derecha c/acarreo	RCR Op,Cantidad	► MSB … LSB ► CF	*i*								±
ROL	Rotar a la izquierda	ROL Op,Cantidad	CF ◄ MSB … LSB ◄	*i*								±
ROR	Rotar a la derecha	ROR Op,Cantidad	► MSB … LSB ► CF	*i*								±

i para más información ver especificaciones de la intrucción   ♦ entonces CF:=0, OF:=0 sino CF:=1, OF:=1

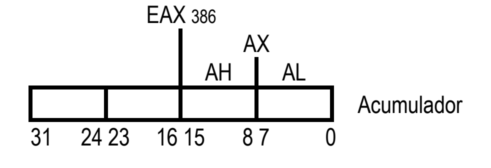
LÓGICOS				Flags								
Nombre	**Comentario**	**Código**	**Operación**	**O**	**D**	**I**	**T**	**S**	**Z**	**A**	**P**	**C**
NEG	Negación (complemento a 2)	NEG Op	Op:=0-Op si Op=0 entonces CF:=0 sino CF:=1	±				±	±	±	±	±
NOT	Invertir cada bit	NOT Op	Op:=Ø Op (invierte cada bit)									
AND	'Y' (And) lógico	AND Dest,Fuente	Dest:=DestÙ Fuente	0				±	±	?	±	0
OR	'O' (Or) lógico	OR Dest,Fuente	Dest:=DestÚFuente	0				±	±	?	±	0
XOR	'O' (Or) exclusivo	XOR Dest,Fuente	Dest:=Dest (xor) Fuente	0				±	±	?	±	0
SHL	Desplazam. lógico a la izq.	SHL Op,Cantidad	CF ◄ MSB … LSB ◄ 0	*i*				±	±	?	±	±
SHR	Desplazam. lógico a la der.	SHR Op,Cantidad	0 ► MSB … LSB ► CF	*i*				±	±	?	±	±

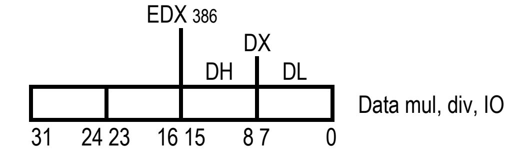
MISCELÁNEOS				Flags								
Nombre	Comentario	Código	Operación	O	D	I	T	S	Z	A	P	C
NOP	Hacer nada	NOP	No hace operación alguna									
LEA	Cargar direciión Efectiva	LEA Dest,Fuente	Dest := dirección fuente									
INT	Interrupción	INT Num	Interrumpe el progr. actual, corre la subrutina de int.			0	0					

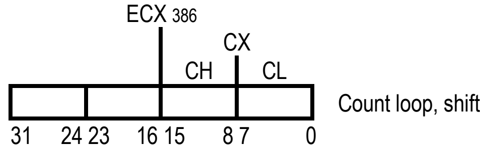
SALTOS (generales)							
Nombre	Comentario	Código	Operación	Name	Comentario	Código	Operación
CALL	Llamado a subrutina	CALL Proc		RET	Retorno de subrutina	RET	
JMP	Saltar	JMP Dest					
JE	Saltar si es igual	JE Dest	(≡ JZ)	JNE	Saltar si no es igual	JNE Dest	(≡ JNZ)
JZ	Saltar si es cero	JZ Dest	(≡ JE)	JNZ	Saltar si no es cero	JNZ Dest	(≡ JNE)
JCXZ	Saltar si CX es cero	JCXZ Dest		JECXZ	Saltar si ECX es cero	JECXZ Dest	386
JP	Saltar si hay paridad	JP Dest	(≡ JPE)	JNP	Saltar si no hay paridad	JNP Dest	(≡ JPO)
JPE	Saltar si hay paridad par	JPE Dest	(≡ JP)	JPO	Saltar si hay paridad impar	JPO Dest	(≡ JNP)

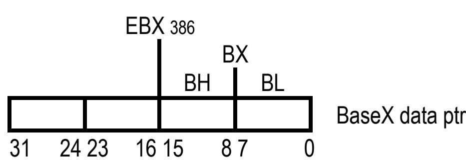
SALTOS Sin Signo (Cardinal)				SALTOS Con Signo (Integer)			
JA	Saltar si es superior	JA Dest	(≡ JNBE)	JG	Saltar si es mayor	JG Dest	(≡ JNLE)
JAE	Saltar si es superior o igual	JAE Dest	(≡ JNB ≡ JNC)	JGE	Saltar si es mayor o igual	JGE Dest	(≡ JNL)
JB	Saltar si es inferior	JB Dest	(≡ JNAE ≡ JC)	JL	Saltar si es menor	JL Dest	(≡ JNGE)
JBE	Saltar si es inferior o igual	JBE Dest	(≡ JNA)	JLE	Saltar si es menor o igual	JLE Dest	(≡ JNG)
JNA	Saltar si no es superior	JNA Dest	(≡ JBE)	JNG	Saltar si no es mayor	JNG Dest	(≡ JLE)
JNAE	Saltar si no es super. o igual	JNAE Dest	(≡ JB ≡ JC)	JNGE	Saltar si no es mayor o igual	JNGE Dest	(≡ JL)
JNB	Saltar si no es inferior	JNB Dest	(≡ JAE ≡ JNC)	JNL	Saltar si no es inferior	JNL Dest	(≡ JGE)
JNBE	Saltar si no es infer. o igual	JNBE Dest	(≡ JA)	JNLE	Saltar si no es menor o igual	JNLE Dest	(≡ JG)
JC	Saltar si hay carry	JC Dest		JO	Saltar si hay Overflow	JO Dest	
JNC	Saltar si no hay carry	JNC Dest		JNO	Saltar si no hay Overflow	JNO Dest	
				JS	Saltar si hay signo (=negativo)	JS Dest	
				JNS	Saltar si no hay signo (=posit.)	JNS Dest	

Registros Generales:









Ejemplo:

```
        .DOSSEG             ; Programa de demostración
        .MODEL SMALL
        .STACK 1024
Two     EQU 2               ; Constante
        .DATA
VarB    DB ?                ; define un Byte, cualquier valor
VarW    DW 1010b            ; define un Word, en binario
VarW2   DW 257              ; define un Word, en decimal
VarD    DD 0AFFFFh          ; define un DoubleWord, en hexa
S       DB "Hello !",0      ; define un String
        .CODE
main:   MOV AX,DGROUP       ; resuelto por el linker
        MOV DS,AX           ; inicializa el reg. de segmento de datos
        MOV [VarB],42       ; inicializa VarB
        MOV [VarD],-7       ; setea VarD
        MOV BX,Offset[S]    ; dirección de "H" de "Hello !"
        MOV AX,[VarW]       ; poner el valor en el acumulador
        ADD AX,[VarW2]      ; suma VarW2 a AX
        MOV [VarW2],AX      ; almacena AX en VarW2
        MOV AX,4C00h        ; regresa al sistema
        INT 21h
        END main
```

KEEP IT SIMPLE, STUPID (AND SHORT) - KISS

Flags: - - - - O D I T S Z - A - P - C

Flags de Control (cómo se manejan las instrucciones):

D: Dirección 1=Los op's String se procesan de arriba hacia abajo
I: Interrupción Indica si pueden ocurrir interrupciones o no.
T: Trampa Paso por paso para debugging

Flags de Estado (resultado de las operaciones):

C: Carry resultado de operac. sin signo es muy grande o inferior a cero
O: Overflow resultado de operac. sin signo es muy grande o pequeño.
S: Signo Signo del resultado. Razonable sólo para enteros. 1=neg. 0=pos.
Z: Cero Resultado de la operación es cero. 1=Cero
A: Carru Aux. Similar al Carry, pero restringido para el nibble bajo únicamente
P: Paridad 1=el resultado tiene cantidad par de bits en uno